

BÁO CÁO ĐỒ ÁN THỰC HÀNH
MÔN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU
GVHD: Tiết Gia Hồng

THÔNG TIN NHÓM

Bảng đóng góp của từng thành viên cho lần 3

STT	MSSV	Họ tên	% Đề án lần 3	% Đóng góp
1	19127017	Trương Gia Đạt	30%	10%
2	19127365	Phan Khải Đông		10%
3	19127401	Lê Trung Hiếu		10%

Bảng phân công công việc

STT	Công việc	Người thực hiện	% Hoàn thành	% Hoàn thành số công việc được giao
1	Viết lỗi unrepeatable read và dirty read	Trương Gia Đạt	100%	100%
2	Viết lỗi lost update và conversion deadlock	Phan Khải Đông	100%	100%
3	Viết lỗi phantom và cycle deadlock	Lê Trung Hiếu	100%	100%

CÀI ĐẶT TÌNH HUỐNG TRANH CHẤP

I. Sinh viên thực hiện: Trương Gia Đạt

1. Tình huống 1: Unrepeatable read

ERROR1: Dirty read T1 (User = tài xế): thực hiện xem tình trạng đơn hàng X. T2 (User = đối tác): thực hiện sửa tình trạng đơn hàng X.			
USP_CAU11_TX	Khóa	USP_CAU11_DT	Khóa
Input: @madonhang CHAR(20), @matx CHAR(10) Output: Tình trạng đơn hàng X của tài xế.		Input: @madonhang CHAR(20), @madt CHAR(10), @tinhttrang INT Output: Tình trạng đơn hàng X của đối tác được cập nhật	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) tài xế IF NOT EXISTS (SELECT * FROM TAIXE WHERE MATX = @matx) BEGIN PRINT 'TÀI XẾ KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(TAIXE) //Xin khoá đọc trên bảng TAIXE với đk MATX = @matx		
B2: Kiểm tra thông tin (2) đơn hàng IF NOT EXISTS (SELECT * FROM DONHANG WHERE MADH = @madonhang) BEGIN PRINT 'ĐƠN HÀNG KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(DONHANG) //Xin khoá đọc trên bảng DONHANG với đk MADH = @madonhang		

<p>B3: Kiểm tra thông tin (3) đơn hàng của tài xế IF NOT EXISTS (SELECT * FROM DONHANG WHERE MADH = @madonhang AND MATX = @matx) BEGIN PRINT N'TÀI XẾ KHÔNG CÓ ĐƠN HÀNG</p>	<p>R(DONHANG) //Không cần xin khoá</p>		
<p>WAITFOR DELAY '00:00:10'</p>			
		BEGIN TRAN	
		<p>B1: Kiểm tra thông tin (1) đối tác IF NOT EXISTS (SELECT * FROM DOITAC WHERE MADT = @madt) BEGIN PRINT N'ĐỐI TÁC KHÔNG TỒN TẠI'</p>	<p>R(DOITAC) //Xin khoá đọc trên bảng DOITAC với đk MADT = @madt</p>
		<p>B2: Kiểm tra thông tin (2) đơn hàng IF NOT EXISTS (SELECT * FROM DONHANG WHERE MADH = @madonhang) BEGIN PRINT N'ĐƠN HÀNG KHÔNG TỒN</p>	<p>R(DONHANG) //Không cần xin khoá</p>
		<p>TẠI' ROLLBACK TRAN RETURN 0 END</p>	

		<p>B3: Kiểm tra thông tin (3) đơn hàng của đối tác</p> <pre> IF NOT EXISTS (SELECT * FROM DONHANG WHERE MADH = @madonhang AND MADT = @madt) BEGIN PRINT N'ĐỐI TÁC KHÔNG CÓ ĐƠN HÀNG TRÊN' ROLLBACK TRAN RETURN 0 END </pre>	<p>R(DONHANG) //Không cần xin khoá</p>
		<p>B4: Cập nhật tình trạng(4) đơn hàng của đối tác</p> <pre> IF @tinhtrang > 3 OR @tinhtrang <= 0 OR @tinhtrang <= (SELECT TINHTRANG FROM DONHANG WHERE MADT = @madt AND MADH = @madonhang) BEGIN PRINT N'CẬP NHẬT TÌNH TRẠNG KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN 0 END </pre>	<p>R(DONHANG) //Không cần xin khoá</p>
		<pre> UPDATE DONHANG SET TINHTRANG = @tinhtrang WHERE MADT = @madt AND MADH = @madonhang </pre>	
		COMMIT	
<pre> SELECT TINHTRANG FROM DONHANG WHERE MATX = @matx AND MADH = @madonhang </pre>			

COMMIT			
--------	--	--	--

2. Tình huống 2: Dirty read

ERROR2: Dirty read T1 (User = tài xế): thực hiện cập nhật tình trạng đơn hàng X. T2 (User = khách hàng): thực hiện xem tình trạng đơn hàng X.			
USP_CAU6_TX	Khóa	USP_CAU6_KH	Khóa
Input: @madonhang CHAR(20), @matx CHAR(10), @tinhtrang INT Output: Tình trạng đơn hàng X của tài xế được cập nhật		Input: @madonhang CHAR(20), @makh CHAR(10) Output: Tình trạng đơn hàng X của khách hàng.	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) tài xế IF NOT EXISTS (SELECT * FROM TAIXE WHERE MATX = @matx) BEGIN PRINT N'TÀI XẾ KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(TAIXE) //Xin khoá đọc trên bảng TAIXE với đk MATX = @matx		
B2: Kiểm tra thông tin (2) đơn hàng IF NOT EXISTS (SELECT * FROM DONHANG WHERE MADH = @madonhang) BEGIN PRINT N'DƠN HÀNG KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(DONHANG) //Xin khoá đọc trên bảng DONHANG với đk MADH = @madonhang		

B3: Kiểm tra thông tin (3) đơn hàng của tài xế IF NOT EXISTS (SELECT * FROM DONHANG WHERE MADH = @madonhang AND MATX = @matx) BEGIN PRINT N'TÀI XẾ KHÔNG CÓ ĐƠN HÀNG TRÊN' ROLLBACK TRAN RETURN 0 END	R(DONHANG) //Không cần xin khoá		
B4: Cập nhật tình trạng(4) đơn hàng của đối tác IF @tinhtrang > 3 OR @tinhtrang <= 0 OR @tinhtrang <= (SELECT TINHTRANG FROM DONHANG WHERE MATX = @matx AND MADH = @madonhang) BEGIN PRINT N'CẬP NHẬT TÌNH TRẠNG KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN 0 END	R(DONHANG) //Không cần xin khoá		
WAITFOR DELAY '0:0:10'			
		BEGIN TRAN	
		B1: Kiểm tra thông tin (1) khách hàng IF NOT EXISTS (SELECT * FROM KHACHHANG WHERE MAKH = @makh) BEGIN PRINT N'KHÁCH HÀNG KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0	R(KHACHHANG) //Xin khoá đọc trên bảng KHACHHANG với đk MAKH = @makh

		END	
		<p>B2: Kiểm tra thông tin (2) đơn hàng</p> <p>IF NOT EXISTS (SELECT * FROM DONHANG WHERE MADH = @madonhang)</p> <p>BEGIN</p> <p>PRINT N'DƠN HÀNG KHÔNG TỒN</p> <p>TẠI</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>R(DONHANG)</p> <p>//Không cần xin khoá</p>
		<p>B3: Kiểm tra thông tin (3) đơn hàng của khách hàng</p> <p>IF NOT EXISTS (SELECT * FROM DONHANG WHERE MADH = @madonhang AND MAKH = @makh)</p> <p>BEGIN</p> <p>PRINT N'KHÁCH HÀNG KHÔNG CÓ ĐƠN HÀNG TRÊN'</p> <p>ROLLBACK TRAN</p> <p>RETURN 0</p> <p>END</p>	<p>R(DONHANG)</p> <p>//Không cần xin khoá</p>
		SELECT TINHTRANG FROM DONHANG WHERE MAKH = @makh AND MADH = @madonhang	
		COMMIT	
UPDATE DONHANG SET TINHTRANG = @tinhttrang WHERE MATX = @matx AND MADH = @madonhang			
COMMIT			

II. Sinh viên thực hiện: Phan Khải Đông

1. Tình huống 1: Lost update

ERROR3: Lost update T1 (User = đối tác): thực hiện cập nhật giá sản phẩm X. T2 (User = đối tác): thực hiện cập nhật giá sản phẩm X.			
USP_CAU14_DT	Khóa	USP_CAU14_DT	Khóa
Input: @masp CHAR(15), @madt CHAR(10), @dongia INT Output: Đơn giá sản phẩm được cập nhật.		Input: @masp CHAR(15), @madt CHAR(10), @dongia INT Output: Đơn giá sản phẩm được cập nhật.	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) đối tác IF NOT EXISTS (SELECT * FROM DOITAC WHERE MADT = @madt) BEGIN PRINT N'ĐỐI TÁC KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(DOITAC) //Xin khoá đọc trên bảng DOITAC với đk MADT = @madt		
B2: Kiểm tra thông tin (2) sản phẩm IF NOT EXISTS (SELECT * FROM SANPHAM WHERE MASP = @masp) BEGIN PRINT N'SẢN PHẨM KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(SANPHAM) //Xin khoá đọc trên bảng SANPHAM với đk MASP = @masp		
B3: Kiểm tra thông tin (3) đầu vào đơn giá IF @dongia < 0			

BEGIN PRINT N'GIÁ KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN 0 END			
WAITFOR DELAY '0:0:10'			
		BEGIN TRAN	
		B1: Kiểm tra thông tin (1) đối tác IF NOT EXISTS (SELECT * FROM DOITAC WHERE MADT = @madt) BEGIN PRINT N'ĐỐI TÁC KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(DOITAC) //Không cần xin khoá
		B2: Kiểm tra thông tin (2) sản phẩm IF NOT EXISTS (SELECT * FROM SANPHAM WHERE MASP = @masp) BEGIN PRINT N'SẢN PHẨM KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(SANPHAM) //Không cần xin khoá
		B3: Kiểm tra thông tin (3) đầu vào đơn giá IF @dongia < 0 BEGIN PRINT N'GIÁ KHÔNG HỢP LỆ' ROLLBACK TRAN	

		RETURN 0 END	
		UPDATE SANPHAM SET DONGIA = @dongia WHERE MASP = @masp	
		COMMIT	
UPDATE SANPHAM SET DONGIA = @dongia WHERE MASP = @masp			
COMMIT			

2. Tình huống 2: Conversion DeadLock

ERROR5: CONVERSION DEADLOCK T1 (User = đối tác): thực hiện sửa giá của sản phẩm T2 (User = đối tác): thực hiện sửa giá của sản phẩm			
SUA_GIA_DOITAC1	Khóa	SUA_GIA_DOITAC2	Khóa
<u>Input:</u> @MaDT CHAR(10), @MaSP CHAR(15), @DonGia INT <u>Output:</u> Sản phẩm được cập nhật giá mới SET TRAN ISOLATION LEVEL REPEATABLE READ		<u>Input:</u> @MaDT CHAR(10), @MaSP CHAR(15), @DonGia INT <u>Output:</u> Sản phẩm được cập nhật giá mới SET TRAN ISOLATION LEVEL REPEATABLE READ	
BEGIN TRAN			
WAITFOR DELAY '0:0:10'			
		BEGIN TRAN	
		B1: Kiểm tra thông tin (1) đối tác IF NOT EXISTS (SELECT* FROM DOITAC WHERE MaDT = @MADT) BEGIN	R(DOITAC) //Xin khoá đọc trên bảng DOITAC với

		PRINT N'ĐỐI TÁC KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN END	đk MADT = @madt
		B2: Kiểm tra thông tin (2) sản phẩm IF NOT EXISTS (SELECT* FROM SANPHAM WHERE MaSP = @MASP) BEGIN PRINT N'SẢN PHẨM KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN END	R(SANPHAM) //Xin khoá đọc trên bảng SANPHAM với đk MASP = @MASP
		B3: Kiểm tra đơn giá sản phẩm IF @DONGIA < 0 BEGIN PRINT N'GIÁ KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN END	
		WAITFOR DELAY '0:0:10'	
B1: Kiểm tra thông tin (1) đối tác IF NOT EXISTS (SELECT* FROM DOITAC WHERE MaDT = @MADT) BEGIN PRINT N'ĐỐI TÁC KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN END	R(DOITAC) //Không cần xin khóa		
B2: Kiểm tra thông tin (2) sản phẩm	R(SANPHAM)		

<pre>IF NOT EXISTS (SELECT* FROM SANPHAM WHERE MaSP = @MASP) BEGIN PRINT N'SẢN PHẨM KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN END</pre>	//Không cần xin khóa		
<pre>B3: Kiểm tra đơn giá sản phẩm IF @DONGIA < 0 BEGIN PRINT N'GIÁ KHÔNG HỢP LỆ' ROLLBACK TRAN RETURN END</pre>			
<pre>UPDATE SANPHAM SET DONGIA = @DONGIA WHERE MASP = @MASP</pre>			
COMMIT			
		<pre>UPDATE SANPHAM SET DONGIA = @DONGIA WHERE MASP = @MASP</pre>	
		COMMIT	

III. Sinh viên thực hiện: Lê Trung Hiếu

1. Tình huống 1: Phantom read

ERROR4: PHANTOM T1 (User = đối tác): thực hiện thay đổi mật khẩu tài khoản. T2 (User = quản trị viên): thực hiện xóa tài khoản đối tác.			
USP_CAU13_DT	Khóa	USP_CAU13_QTV	Khóa
Input: @madt CHAR(10), @mkhientai VARCHAR(10), @mkmoi VARCHAR(8) Output: Mật khẩu mới được cập nhật.		Input: @madt CHAR(10) Output: Tài khoản đối tác bị xóa khỏi csdl.	
BEGIN TRAN			
B1: Kiểm tra thông tin (1) đối tác IF NOT EXISTS (SELECT * FROM DOITAC WHERE MADT = @madt) BEGIN PRINT N'DỐI TÁC KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(DOITAC) //Xin khoá đọc trên bảng DOITAC với đk MADT = @madt		
B2: Kiểm tra thông tin (2) tài khoản IF NOT EXISTS (SELECT * FROM TAIKHOAN WHERE MATK = @madt AND MATKHAU = @mkhientai) BEGIN PRINT N'SAI MẬT KHẨU' ROLLBACK TRAN RETURN 0 END	R(TAIKHOAN) //Xin khoá đọc trên bảng TAIKHOAN với đk MATK = @madt AND MATKHAU = @mkhientai		
WAITFOR DELAY '0:0:10'			

		BEGIN TRAN	
		B1: Kiểm tra thông tin (1) đối tác IF NOT EXISTS (SELECT * FROM DOITAC WHERE MADT = @madt) BEGIN PRINT N'ĐỐI TÁC KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN 0 END	R(DOITAC) //Không cần xin khoá
		DELETE TAIKHOAN WHERE MATK = @MADT	
		COMMIT	
UPDATE TAIKHOAN SET MATKHAU = @MKMoi WHERE MATK = @MaDT			
COMMIT			

2. Tình huống 2: Cycle DeadLock

ERROR6: CYCLE DEADLOCK T1 (User = đối tác): xóa chi nhánh T2 (User = đối tác): xóa chi nhánh			
XOA_CHI_NHANH_TRAN1	Khóa	XOA_CHI_NHANH_TRAN2	Khóa
<i>Input:</i> @MACHINHANH CHAR(3), @MADT CHAR(10) <i>Output:</i> Chi nhánh bị xóa khỏi hệ thống SET TRAN ISOLATION LEVEL REPEATABLE READ		<i>Input:</i> @MACHINHANH CHAR(3), @MADT CHAR(10) <i>Output:</i> Chi nhánh bị xóa khỏi hệ thống SET TRAN ISOLATION LEVEL REPEATABLE READ	
BEGIN TRAN			

B1: Kiểm tra thông tin (1) chi nhánh IF NOT EXISTS (SELECT * FROM CHINHANH WHERE MACN = @MACHINHANH AND MADT = @MADT) BEGIN PRINT N' CHI NHÁNH NÀY KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN END	R(CHINHANH) // Xin khoá đọc trên bảng CHINHANH với đk MACN = @MACHINHANH và MADT = @MADT		
UPDATE HOPDONG SET SOCHINHANH = SOCHINHANH - 1 WHERE MAHD = (SELECT MAHD FROM CHINHANH WHERE MACN = @MACHINHANH AND MADT = @MADT)			
WAITFOR DELAY '0:0:10'			
		BEGIN TRAN	
		B1: Kiểm tra thông tin (1) chi nhánh IF NOT EXISTS (SELECT * FROM CHINHANH WHERE MACN = @MACHINHANH AND MADT = @MADT) BEGIN PRINT N' CHI NHÁNH NÀY KHÔNG TỒN TẠI' ROLLBACK TRAN RETURN END	R(CHINHANH) // Không cần xin khóa đọc
		DECLARE @MAHD AS CHAR(10)	

		SET @MAHD = (SELECT MAHD FROM CHINHANH WHERE @MACHINHANH = MACN AND MADT = @MADT)	
		DELETE CHINHANH WHERE MACN = @MACHINHANH AND @MADT = MADT	
		UPDATE HOPDONG SET SOCHINHANH = SOCHINHANH - 1 WHERE MAHD = @MAHD	
		COMMIT	
DELETE CHINHANH WHERE MACN = @MACHINHANH AND MADT = @MADT			
COMMIT			